eMeeting.net
Phase 2 Databases
Initial author: Trey Smith

Index numbers are the raw, zero-based Btrieve index/key numbers. With ZBASE2/BDAC2, these are one-based. So index 0 here is index 1 to BDAC2.

Notice most of the databases accessed via ODBC have an autoinc as a unique (duplicates not allowed) index. This keeps ODBC programs from choking. Btrieve automatically supplies a "system" index if a unique index is not supplied, but ODBC doesn't seem to recognize it. You must have a unique key as key 0. Name this field "recnum".

Note that ZBASE2/BDAC2 combines consecutive date and time fields together into one field internally. That throws your field count from script off unless you're aware of it.

Please note this doc uses the term "database" sometimes to mean a specific table (in the ODBC sense of the word). Sorry if that causes confusion.

BILLING.BTX
recnum - autoinc 4 - index 0
conf_id - int 4 - index 1
cust_id - int 4 - index 2
user_id - int 4 Blank unless this conf requires user ids.
mach_id - char 2
line - char 4
start_date - date
start_time - time
stop_date - date
stop_time - time
duration - int 4
attribs - char 20

Attribs used:
A = Telephone user.
B = IP user.
The system generates one record per participant per conference.

ADDRESS.BTX
cust_id - int 4
user_id - int 4
name1 - char 20
name2 - char 20
home_phone - char 20
work_phone - char 20
email - char 20
misc1 - char 20
misc2 - char 20

Index 0: cust_id + user_id
index 1: cust_id Allows program to display the complete address book for a customer.

User_id 0 is the owner's address book entry for himself; user_id 0 is entered automatically by the system when the user registers. The user_id field links to a next_user_id field in the customer record. Tango uses that next_user_id field to populate the user_id field upon creation of new

address book entries. The next_user_id field is thus a per customer code-incremented counter providing a unique user_id for each address book entry for a particular cust_id.

DISPLAY.BTX
recnum - autoinc 4 - index 0
conf_id - int 4 - index 1
user_id - int 4
name1 - char 20
name2 - char 20
attribs - char 20
timeslot - int 4 - index 2

Attribs used:
A = Half duplex (muted) caller.

C_USER.BTX
recnum - autoinc 4 - index 0
conf_id - int 4 - index 1
user_id - int 4
name1 - char 20
name2 - char 20
attribs - char 20
timeslot - int 4 - index 2

Attribs:
B: Set when a user has been added to a conference and is checked so that the user will not receive another conference-confirmation email.
E: Set when a user has been added to a conference as default. If "E" is present, then the user will receive an email message about the conference, else user will not receive an email
P: Set when a user has been added to a conference as default. If "P" is present, then the user will designated as a talk/listen, else user will be set as just listen.

CONF_ID.BTX
conf_id - autoinc 4 - index 0
cust_id - char 20 - index 1
phone_num - char 20
max_users - char 3
start_date - date
start_time - time
stop_date - date
stop_time - time
duration - char 4
attribs - char 20
pass_talk - char 20
pass_listen - char 20

The attribs field holds:
A: Set when conference is 'Advance' and thus the user will be prompted to enter a user_id.

RTCONF.BTX
conf_id - int 4 - index 0
realconf - int 2 - index 1
curusers - int 2

realconf is the actual conference number used by the conference engine. If no rtconf record for a conf_id exists, the voice node does a seek_last on this field, increments the value it finds, uses that new value to insert a new record for its conf_id. If the rtconf database is empty, the voice node inserts a record for the conf_id with realconf (and curusers) set to 1.
curusers increments as users join the conference.

CALENDAR.BTX
CALENDAR.DLL uses CALENDAR.BTX directly via Btrieve API calls. There is no applicable ODBC or BDAC2 definition for this database. Do not attempt to edit or modify this database via DDF Ease, Access, or any other method. The Tango application uses CALENDAR.DLL to perform resource allocation.

Database fields:
date - date
segments - char 192 - A single two byte short int per segment.

These two fields relate directly to the C type:

```
{
btrieve_date_type date;
unsigned short int segments[SEGMENTS_PER_DAY];
}
```